Indirect sensing through abstractive learning.

Article (Unspecified)

Thornton, Chris (2003) Indirect sensing through abstractive learning. Intelligent Data Analysis, 7 (3). pp. 255-266. ISSN 1088-467X

Indirect Sensing Through Abstractive Learning

Chris Thornton
Cognitive and Computing Sciences
University of Sussex
Brighton
BN1 9QH
UK

Email: Christopher.Thornton@firenet.uk.com
Tel: (44)1273 678856

May 21, 2003

Abstract

The paper discusses disparity issues in sensing tasks involving the production of a 'high-level' signal from 'low-level' signal sources. It introduces an abstraction theory which helps to explain the nature of the problem and point the way to a solution. It proposes a solution based on the use of supervised adaptive methods drawn from artificial intelligence. Finally, it describes a set of empirical experiments which were carried out to evaluate the efficacy of the method.

KEYWORDS: abstraction, sensing, learning, prediction

1 Introduction

Sensing technologies have advanced rapidly in the last few decades [1]. But it remains the case that the phenomena we are most likely to be interested in are precisely those for which the sensing task is most challenging. For example, we can easily construct a sensor that will automatically trigger a door opening mechanism when a large, warm object approaches. But it is a greater challenge to make the sensor open the door only if the large, warm object is a member of a particular family.

The salient phenomena of our world tend to be macroscopic objects and processes — people, especially, and the objects and processes they interact with. Sensing technologies, on the other hand, interface to the most basic, physical level of reality — levels of light, heat and motion. This sets the scene for the so-called *disparity* problem. We would like to be able to sense *high-level* properties and objects of our world, because this is where our interests lie. But the technologies on offer provide access to characteristically *low-level* properties. Early texts anticipating disparity issues include [2, 3, 4]. [5] is also of interest.

This problem with the technology looks as if it should be easily overcome. High-level phenomena must impact low-level phenomena in some specific way, i.e., they must create specific signatures. So we might expect to be able to build systems that will use these signatures to identify the relevant, higher level objects within the low-level data. The deficiencies in sensing technologies should then be curable using signal processing. But the question is, what is this signal processing to consist of? And how is it to be accomplished?

One idea is to make use of supervised, adaptive methods. Such methods are commonly used in areas such as artificial intelligence [6], machine learning [7], digital signal processing [8], statistics [9], data mining [10], neural networks [11] and genetic algorithms [12]. They aim to produce a mechanism, formula or theory capable of generating appropriate outputs from arbitrary inputs. They do this using a reference set of input/output examples (often called a *training set*).

As a response to the disparity problem, supervised, adaptive methods are attractive, since they promise a solution without the need for hands-on design. Whether or not they can fulfil this promise is an interesting question. The present paper examines the situation using a combination of theoretical analysis and empirical experiment. Building on ideas previously put forward in [13] and [14] it outlines an *abstraction theory*, which offers a computational analysis of the disparity problem. It then explores a solution to the problem using *indirect sensing* based on an adaptive method from the area of computational learning. Finally, it presents the results of empirical evaluations of three different adaptive methods applied to a challenging disparity task.

2 Abstraction theory

The preliminary task for the paper is to set out a framework in which we can identify the origins of the disparity problem and understand why its impact is problematic. Sensing is a procedure which attempts to produce a signal from an environment. So we start the process by considering the basic question: What is an environment?

Anything that we would wish to honour with the name must be based on some set of basic elements. And these should be viewed as the primitive phenomena of the environment. The majority of phenomena, however, will most likely consist of combinations of these basic elements. But there are two, quite different ways in which phenomena may be combined together to form new entities. First, there is the process of *composition* in which several elements are combined together to form a new whole. Second, there is the process of *classification* by which several elements are gathered together into a single class.[1] Every possible subgroup of basic elements is a candidate for both processes.

[1] The two forms of combination are known to artificial intelligence researchers as 'partof' and 'isa' construction [15].

Thus, starting from the set of basic phenomena, we may derive a new level of potential phenomena by treating each possible subgroup as (a) a composite and (b) a class.

The general idea is visualised in Figure 1. Here the basic level of phenomena is visualised as a set P_0 of entities (the grey circles are meant to represent individual phenomena). From P_0, we obtain P_1: each phenomenon here is derived by applying composition or clustering to a subset of P_0. Treating P_1 as the set of basic phenomena now permits the derivation of a set P_2 in which each phenomenon is the result of clustering or composition applied to a subset of P_1. Treating P_2 as the set of basic phenomenon permits the derivation of the set P_3 and so on. In this manner, we can go on to derive P_4, P_5, P_6 etc.

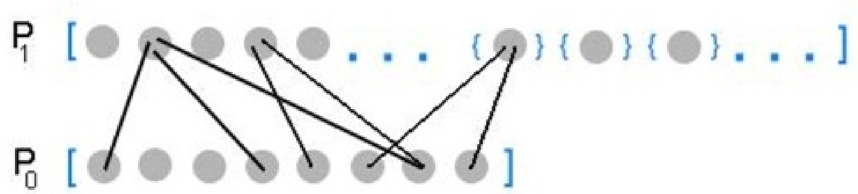

Figure 1:

Note that in the case of both composition and clustering, a set of elements is reidentified as a single entity, with an accompanying elimination of information relating to independent elements. In the case of composition, the elements become the component parts of a new whole. In clustering, the elements become alternative manifestations of a single identity. These processes are thus quintessentially *abstractive*. The tree of derived phenomena — a small fragment of which appears in Figure 1— is thus a tree of abstractions: an *abstraction tree*.

The vertical dimension of the tree provides a visual interpretation for what is meant by 'high-level' and 'low-level'. In terms of the abstraction tree, high-level phenomena are phenomena appearing in the upper layers of the abstraction tree while low-level phenomena are phenomena appearing in the lower layers.

3 Complexity

Applied recursively to a set of basic phenomena, the two forms of abstraction lead to an infinite hierarchy of potential phenomena. The number of nodes in this hierarchy expands rapidly as we move upwards from level to level. Let us say there are n nodes in a particular layer. The layer above adds two instantiations of each subset. Thus, the number of nodes it contains must be $2^n + 2^n$.

If we label the set of nodes at the i'th level of the hierarchy P_i, and let $P_o = n$, the number of nodes at any given level of the hierarchy is thus

$$|P_i| = 2^{(|P_{i-1}|+|P_{i-1}|)} - |P_{i-1}|$$

The subtraction of $|P_{i-1}|$ in the recursive equation is necessary in order to discount the fact that singleton subsets resulting from composition are identical to singleton subsets resulting from clustering. We can offset this effect by discount exactly one set of singleton subsets.

4 Poker hands example

To make these ideas more concrete, let us take, as our basic environment (i.e., our set of basic elements) a pack of playing cards. Standard poker hands may then be viewed as phenomena of different levels of abstraction.

At the first and lowest level of abstraction we have classes and compositions constituted from the actual playing cards. An example of a class would be 'heart', i.e, the class of all hearts. Another example would be 'two' — the class of all twos. We also have composites, i.e., collections of specific cards. An example would be the pair consisting of the two of spades and the nine of hearts.

At the second level of abstraction we have classes and compositions made up from first level entities. An example of a class at this level is 'two-of-a-kind': this contains all two-card hands from the first-level in which both cards have the same rank. For similar reasons, 'three-of-a-kind' is also a class at this level, as is 'straight', 'flush', 'four-of-a-kind', 'straight-flush' and 'royal-flush'. Specific hands *within* these categories constitute second level compositions.

At the third level of abstraction we have classes and composites made up from second-level entities. An example of a composition at this level would be a specific full-house, i.e., an entity combining a specific two-of-a-kind with a specific three-of-a-kind.

At the fourth level of abstraction, the classes and compositions are made up from third-level entities. And this is the level at which we naturally identify the full-house *class*, i.e., the class of all full-houses.

Above this level, the potential entities are no longer defined within conventional poker classifications. They can thus be viewed as potential constructs which are, in fact, non-existent within the domain of poker.

5 Types and tokens

Mapping out the generative properties of abstraction sheds some light on the relationship between types and tokens. Any phenomenon in the abstraction tree whose derivation is not, at any stage, mediated by clustering (i.e., whose roots do not go back through any class nodes) has only one, possible grounding in basic elements — there is only way it can possibly occur or exist. In contrast, any phenomenon whose derivation *has* been mediated by clustering (i.e., whose roots do go back through class nodes) is a phenomenon with more than one possible grounding in basic elements. The latter type of phenomenon is what we would conventionally call a *type*, since it effectively 'stands for' more than one pattern of basic-level phenomenon. The former type of phenomenon is a *token* by the same reasoning.

When we map out the hierarchy of potential phenomena resulting from abstraction processes, we thus obtain both types and tokens. The differentiation between the two is more a matter of degree than dichotomy. The situation appears to be that every 'thing' has a class-like nature, the degree of which is fixed by the number of basic-level groundings that can be realised.

6 Real and potential entities

Abstraction, then, is capable of generating an infinite hierarchy of potential phenomena. But we would like to know what phenomena really *exist* in specific domains. In particular, we would like to be able to pick out a domain and enumerate the phenomena. Provided we can identify the set of basic elements, we can map out the abstraction tree of potential phenomena. But presumably only a finite fraction of this infinite set will be realised. How do we distinguish the real from the potential?

Whatever domain we choose to look at, we have to assume that the physics of the domain, or the dynamics of the domain, or the God of the domain, or *whatever* it is that causes the domain to be the way it is, will tend to 'put elements together' in a certain way. We cannot say very much about what this 'way' is, without first understanding the relevant physics, dynamics or God. However, one deduction can be made with respect to the composites of the domain. Regardless of what the parts of a particular composite actually are, we know that they must at least 'fit together', i.e., they must have the right relationships to enable them to combine together to form the relevant whole. Composites are thus relational structures by definition.

The implication is that the linkage between basic-level phenomena and higher-level phenomena of the domain may be mediated by relational composites. In fact, this turns out to be guaranteed. Clustering applied solely to classes is a redundant operation — the same effect can always be achieved simply by deriving bigger classes in the first place. Thus we know that linkages spanning more than two layers of any non-redundant abstractions are necessarily mediated by composites. By the same argument, linkages between basic-level and high-level phenomena are also always mediated by relational composites.

This has a bearing on the disparity problem and on the chances of solving it using supervised adaptive methods. Recall that in a disparity sensing task, we require a sensor which responds in some specific way to a high-level phenomenon of an environment on the basis of measures or signals of low-level phenomena. In signal-processing terms, this task involves the creation of a function which maps from basic elements, to high-level outputs (instantiations of high-level phenomena). The abstraction theory enables us to analyse what this is going to involve in different situations.

First of all, there is the trivial case to consider. This occurs when 'high' and 'low' are really the same and the phenomenon to be sensed is actually a basic-level feature of the environment. For example, the 'high-level' phenomenon might be a particular temperature and the low-level data might be temperature readings. In this case, the phenomenon of interest is a level 0 phenomenon and the only signal processing required is that a particular input (element) should trigger a particular output.

Consider now a level 1 class. Sticking with the temperature example, the class might contain a set of temperature values. The signal-processing then requires that any one of a class of inputs (elements) triggers a particular output.

Finally, consider a level 2 class containing level 1 composites. The high/low linkage is now mediated by a relational composite and the required signal processing is more complex. The requirement is that a particular output is produced if basic inputs (elements) can be grouped together to form a composite which is within the relevant class.

How well should we expect supervised, adaptive methods to fare on these three, sensing tasks?

The answer with respect to the trivial case is readily obtained. Any set of input/output examples constructed for this phenomenon will exhibit a completely reliable relationship between specific input values and specific output values. Any supervised method may be expected to exploit this relationship and to produce a reliable mechanism for performing the desired sensing.

The answer with respect to the level 1 phenomenon is similar, except, in this case, the relationship will be between *sets* of input values and particular output values. Again, any supervised method will perform to satisfaction and derive the desired sensing mechanism [13].

However, when we come to consider the level 2 phenomenon, and in fact *any* phenomenon above this level, the situation changes in a significant way. The level 2 phenomenon is necessarily based on a relational composite. The relationship required by the supervised methods now exists between an output value and sets of input values exhibiting particular relationships. The impact of this on the outlook for supervised learning is firmly negative. No longer is there any expectation that there will be reliable relationships between output values and specific input values. In fact the reverse is expected. The individual input values that we will see associated with specific output values are now effectively *unpredictable*, in the absence of information about the related values.

As a result, a set of reference examples constructed for this sensing task will have the appearance of *random* data and chance-level performance is the best that will be achieved by any conventional, adaptive method. The only escape is for the adaptive method to reconstitute the relevant relational composite and then use it as a 'filter' for pre-processing the data. If, as we would expect, the results of this exhibit the required, relationship with output values, supervised learning may proceed as normal.

The general lesson is that relationality is a 'spanner in the works' for conventional, supervised methods. The existence of relationality in the underlying structure of the target phenomenon implies that any reference examples will not show reliable relationships between input and output values. Rather, they may appear to be derived from a random distribution. To overcome this problem, a supervised method must somehow reconstitute the relevant relational aggregate(s) and then use it/them as a filter for pre-processing the data.

7 Introduction to the DISTAL method

This section introduces the DISTAL method, which aims to meet the requirements noted above. Based on an artificial intelligence method called *abstractive learning*, it uses informed search to seek out the relevant structure underlying a set of reference examples for a disparity task. It achieves this by recursively exploring the abstractive constructions that can be placed upon primitive phenomena until such time as a structure is identified which provides a match against the reference examples. In carrying out the search, it takes into account the full taxonomy of phenomena identified in abstraction theory.

7.1 Abstractive learning algorithm

The abstractive learning procedure forms the main component in the DISTAL method. The main steps of the algorithm are as follows.

(1) Read in the set of reference, input/output examples.

(2) Generate the least complex, abstraction structure.

(3) Compute the frequency with which the structure correctly evaluates to the output of a training example when it is applied to the associated inputs (elements).

(4) Save the structure if this frequency is the highest identified so far.

(5) If the complexity of the structure remains within bounds, identify the next most complex structure and continue from step 3.

(6) Output the most recently saved structure.

Unfortunately, searching over structures in this way carries an exponential cost. In the experiments, this was kept at bay by having the implementation focus on a small subspace of the total set of possible structures. This comprised just those structures which could be built from two-part composites, using a set of four relationships. Class phenomena were not explored directly. Instead, every relational composite was assumed to be a member of a class in the level above, i.e., every relational composite was treated as implicitly identifying the class of all composites whose parts exhibited the same relationships.

The relationships actually utilised in the experiment were known as *Sum*, *Diff*, *Sum2* and *Diff2*. The *Sum* relationship evaluates to the sum of its arguments while the *Diff* relationship evaluates to the difference between its arguments. The *Sum2* and *Diff2* relationships are variations of *Sum* and *Diff* which apply a modulus filter to normalise their values within the range of possible target outputs, in this case the range 0..6. These particular relationships were selected for their generality and computational simplicity.

The exploration of possible structures carried out is most easily conceptualised as a bottom-up process. The primitive variables of the environment form the set of basic elements. The first level of possible structures is explored by searching through the possible relationships and possible assignments of relational roles to basic elements. For each configuration, a new level of basic elements may be obtained. These are the outputs (i.e., values) of the applied relationships. The search process then explores the structures that may be erected with respect to these elements just as it did with the original elements.

7.2 How abstractive learning fits into DISTAL

The DISTAL method manages the application of abstractive learning to the relevant sensing problem. It is effectively a 'wrapper' for abstractive learning. The three main steps are as follows.

(1) Acquire the set of reference, input/output examples for the sensing task at hand. These must associate input values with target sensor outputs.

(2) Run the abstractive learning procedure to find the structure which most accurately fits the training data.

(3) Use the structure identified as a way of implementing a sensor/signal-processing mechanism which actually performs the sensing task.

The final product of this procedure is a method for indirectly sensing a high-level property on the basis of low-level inputs.

8 The head-count experiment

To evaluate the DISTAL method as a solution to the disparity problem, an experiment was performed to test its performance on a suitably configured head-counting task. Head-counting involves counting the number of persons passing through a particular area — normally a corridor or passage. The aim in this particular version of the task was to use audio data derived from a microphone feed to detect the passage of individuals up or down a busy staircase. The task thus involved using a low-level sensory signal to derive a measure of a high-level property (the head-count).

To facilitate testing, a large collection of real-world datasets were obtained. Each of these contained samples recording the association between particular patterns of audio input and the passing/non-passing of individuals through the relevant physical space (the staircase). The DISTAL method was tested for its ability to predict the passing/non-passing attribute using a set of samples as a reference.

The data used in the experiment were derived using the digital recording facilities on a standard PC. A microphone was placed on a busy staircase on the University of Sussex campus and connected to the PC. While the signal from the microphone was digitally recorded, an electronic log was kept marking the time at which each person using the staircase passed the microphone point. The recording and the log were then integrated so as to produce a dataset based on six input variables and one output variable, i.e., a set of data in which each datum consisted of seven values, six of them input values and one of them an output value.

Each input variable was the normalised, mean amplitude recorded over a one-second interval and the six input values of each datum were arranged in chronological order to represent a sequence of amplitudes for a six-second interval. The output variable for each datum was set to the number (integer) of individuals passing the microphone point in the relevant interval (up to a maximum of six). 16 recordings were made in all, each one lasting for 18 minutes. From these recordings, 16 datafiles were produced, each one containing 180 data (18 x 60 / 6 = 180).

These datasets may be visualised as simple time series by concatenating the input values (mean amplitudes) from successive data, ignoring the output value. Applying this process to the 'Da' dataset (see below) produces the visualisation shown in Figure 2. This has the expected appearance of a random signal.

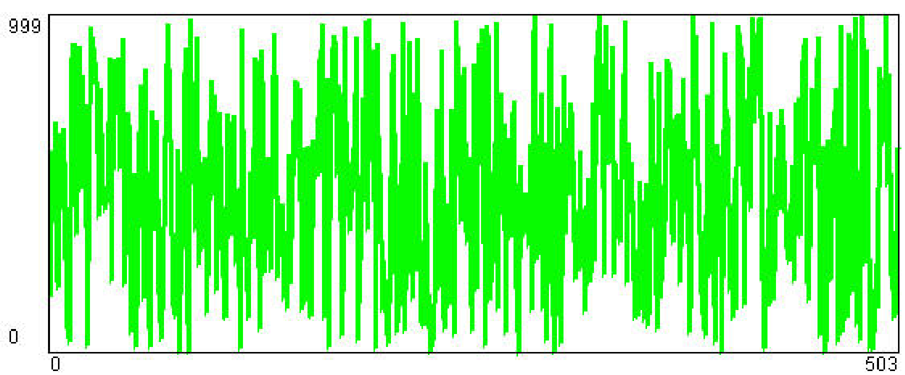

Figure 2:

9 Testing

The empirical evaluation involved applying the DISTAL method along with two other popular adaptive methods to the entire suite of 16 training sets. In each application, the method to be tested was provided with 120 training examples selected randomly from one of the 16 datasets. The method was then tested for its ability to predict the output values (i.e., the number of individuals passing the mic point in the relevant interval) in the remaining 60 samples from the dataset. The mean accuracy on the unseen data was then computed and tabulated.

The other two methods tested were the decision-tree method C4.5 [16] (closely related to the CART method of [17]) and the nearest-neighbor method [18]. The nearest-neighbour method was configured to use six neighbours during the output-generation phase, a setting which proved to generate optimal performance on this task. The C4.5 method was run in the default configuration using the developer's implementation.

Performance statistics were generated by averaging the performance of each method over 100 applications on each dataset, each of which would have provided the method with a different, random selection of training examples. In other words, 100 x 16 x 4 = 6400 test runs were performed in all.

10 Results

The results produced are shown in Figure 3. The vertical axis here represents accuracy: this is the proportion of unseen, test cases on which the method correctly predicted the output value (i.e., correctly predicted the number of individuals passing the microphone point in the relevant time interval). The horizontal axis ranges over the possible datasets, here labelled `Da`, `Db`, `Dc` etc. The results are also shown in tabular form in Table 1.

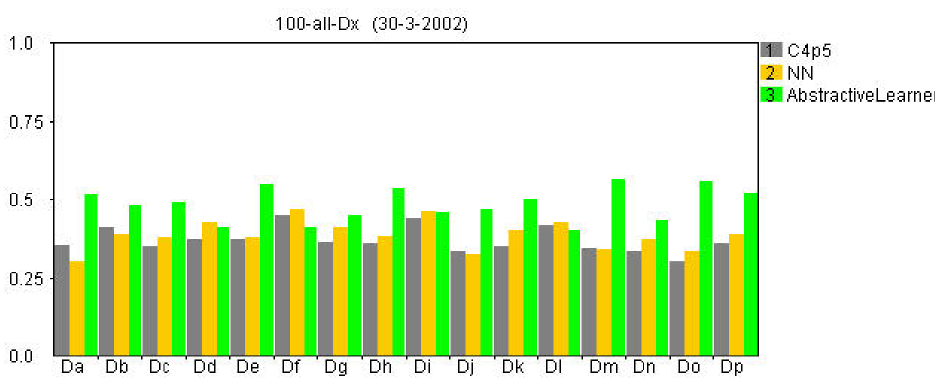

Figure 3: Prediction accuracies for head-count datasets.

	C4p5	NN	DISTAL
Da	0.352	0.298	0.514
Db	0.411	0.385	0.482
Dc	0.348	0.375	0.489
Dd	0.371	0.422	0.411
De	0.372	0.377	0.548
Df	0.449	0.466	0.410
Dg	0.360	0.410	0.448
Dh	0.357	0.382	0.534
Di	0.436	0.46	0.457
Dj	0.333	0.326	0.468
Dk	0.349	0.400	0.502
Dl	0.416	0.424	0.401
Dm	0.344	0.339	0.56
Dn	0.335	0.372	0.435
Do	0.299	0.331	0.555
Dp	0.359	0.384	0.520
M	0.368	0.384	0.483

Table 1:

The final row of the table (row M) gives the mean accuracies achieved. Note that abstractive learning is shown as achieving a mean accuracy of 0.483. This compares favourably against the accuracies achieved by the other two methods, these being 0.384 for the nearest-neighbor method and 0.368 for the C4.5 method. Although the accuracies seem rather low, it must be remembered that target outputs in these datasets were within the integer range 0..6. Thus, the level of accuracy achieved by random guessing would be $1/7 \times 100$, i.e., just over 14%. The accuracy achieved by abstractive learning exceeds this level by some 34%.

11 Performance analysis

The nearest-neighbour method operates by selecting from its store of reference examples the closest match to any unclassified ('unseen') case. The output from the matching example is then produced as the predicted output for the unclassified case. Where the NN method is configured to use multiple nearest-neighbors, this final step involves deriving an average output from the neighbours. Since, the algorithm does little more than store the training data and then utilise it as a lookup table, there is little to be said in the way of performance analysis.

C4.5 operates by building a decision tree for output prediction from the given samples, utilising an information theory measure in order to minimise the depth of the tree (i.e., the average number of decisions that have to be made in order to generate a given output). Following any run of the method, it is possible to look at the decision tree which has been generated and also at the level of generalisation. This corresponds to the average number of cases which are handled by each leaf node.

Unfortunately, the decision trees generated by C4.5 for these datasets are large and provide a minimal degree of generalisation: on average each tip node covers only two training examples. C4.5 appears to react to this domain by creating what are in effect large memories for the training data. This approach is similar to that taken by the nearest-neighbor method so it is not surprising that the two methods produce very similar levels of performance. A small sample of a decision tree generated by C4.5 for the Da dataset is shown in Figure 4.

```
2 <= 571 :
|   0 <= 461 :
|   |   2 <= 122 :
|   |   |   0 <= 240 : 3 (2.0)
|   |   |   0 > 240 : 2 (2.0/1.0)
|   |   2 > 122 :
|   |   |   1 <= 253 :
|   |   |   |   2 <= 416 : 1 (4.0/2.0)
|   |   |   |   2 > 416 : 4 (2.0/1.0)
|   |   |   1 > 253 :
|   |   |   |   0 <= 114 : 1 (3.0/2.0)
|   |   |   |   0 > 114 :
|   |   |   |   |   5 > 411 : 2 (7.0)
|   |   |   |   |   5 <= 411 :
|   |   |   |   |   |   5 <= 347 : 2 (5.0/1.0)
|   |   |   |   |   |   5 > 347 :
|   |   |   |   |   |   |   1 <= 650 : 4 (4.0/1.0)
|   |   |   |   |   |   |   1 > 650 : 6 (2.0/1.0)
```

Figure 4: Initial branches of C4.5 decision tree.

12 Abstractor cascades

Unlike C4.5 and NN, DISTAL achieves a high level of generalisation on these datasets, constructing quite compact 'theories' for the data. Because all the input/output examples showed outputs in the integer range 0..6, DISTAL inevitably selected structures whose top (root) node was either the *Sum2* or the *Diff2* relationship, since only these produced values in the appropriate range (0..6). However, it was observed that the method would produce very different structures for the same data, depending on the selection of cases used for the training sample. Some of the structures which were identified by DISTAL during the experiment are shown in Figure 5. The internal-node labels in this diagrams name the relevant relationship while the tip-node labels show input-variable indexes.

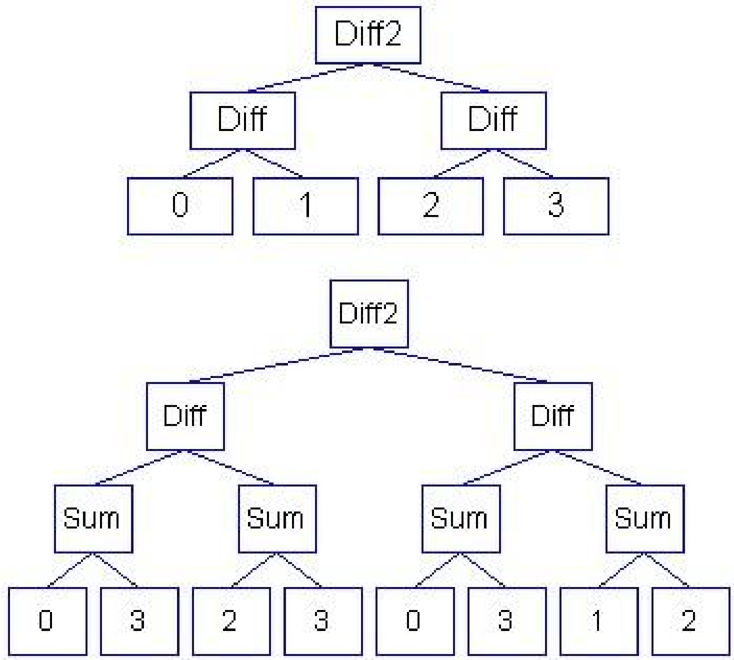

Figure 5: Misc. abstraction cascades for head-count data.

13 Performance comparisons

The head-counting task was explicitly designed as a disparity, sensing task, i.e., a task involving the production of a high-level signal from low-level inputs. Abstraction theory suggests that such tasks will necessarily introduce relational effects and that the net result for conventional supervised adaptive methods will be poor performance. It also suggests that the result for a properly configured, pre-processing method (such as DISTAL) should be good or perfect performance. In the event, the conventional methods performed better than expected while the DISTAL method performed not as well.

The relatively poor performance of the DISTAL method can be put down to a number of factors. First, it may well be that the data obtained for the head-counting task did not have the right characteristics, i.e., they did not reflect a disparity problem of the anticipated type. The characterisation of the task suggests that the output signal relates to a high-level property of the domain while the inputs relate to low-level properties. But this assumption is not made on a formal footing. Although the data may have embodied a disparity problem, it may not have been of a type which was within the search-space explored by the DISTAL method. It may also have been that noise and other unwanted statistical effects have obscured the relevant abstraction structures within the data.

The better than expected performance of the two conventional implies that, at the very least, the data acquisition process resulted in a number of 'impurities'. Neither of the conventional methods tested perform any kind of abstractive pre-processing. On the face of it, they should therefore not have performed above the level of random guessing. The fact that they did perform above this level implies that the derivation of the data must have introduced useful *incidental effects* [13], i.e., fortuitous statistical patterns allowing the prediction of output values on the basis of independent input values. A simple example would involve the statistical prevalence of a particular head-count value associated with an input value (a mean amplitude) falling above a particular threshold. On the other hand, the fact that the decision trees produced by the C4.5 method were so large seems to imply that whatever these incidental effects were they, none of them can have been so pronounced as to allow for the derivation of anything approximating a reliable input/output rule.

14 Review

The paper has focussed on the so-called disparity problem — the non-availability of sensing technologies for precisely those phenomena we expect to be significant in everyday applications.

Abstraction theory explains and quantifies the derivation of phenomena from a set of basic elements. It shows how the level of a particular phenomenon can be quantified by establishing its level within the domain's abstraction tree and how types and tokens are fundamentally equivalent forms of phenomena. It also reveals that high-to-low linkages will always be mediated by relational composities and shows that this has a negative impact on the chances of deriving low-to-high functions using conventional supervised methods.

Empirical experiments were carried out to investigate the idea of using supervised, adaptive methods in order to derive low-to-high signal processing functionality automatically. Attention focussed particularly on the DISTAL method. This yielded prediction accuracies close to the 50% level. Tests using other supervised methods yielded slightly lower but still promising levels of accuracy.

15 Directions for further work

In the experiments performed, the DISTAL method was configured to search for a *single*, cover-all structure. However, this may not be the best approach. It may be that a more fruitful strategy is to let the algorithm find a set of structures and then select from them dynamically when attempting to predict a target output. This would obviously raise challenging complexity issues. However, it would also considerably expand the pre-processing range of the method.

There may also be advantages in relaxing some of the restrictive assumptions concerning the relationships and role attachments. One might allow the algorithm to explore structures involving relationships using more than two roles, for example. One might also allow the algorithm to investigate structures in which more than one role may be attached to the same element, a possibility which is explicitly disallowed in the current implementation. Which, if any, of these modifications will bear fruit remains to be seen. It is hoped that the results of further experiments will form the subject for future publications.

16 Conclusion

The paper has introduced a general theory for analysing the disparity problem, and a method based on adaptive, supervised learning for solving it. The initial results show a performance level considerably below that which would be required for serious applications work. However, results using the two, conventional supervised show some promise in that they considerably exceed the level of performance that would be expected from random guessing. The performance level achieved by the DISTAL method is slightly better than that achieved by the two, conventional methods and might therefore be viewed as more promising still. Future work will pursue a much broader empirical study in the hope of clarifying the issue.

References

[1] Gibson, P. (2000). *Introductory Remote Sensing: Principles and Concepts.* Routledge Publishers.

[2] Bowley, A. (1907). *Elements of Statistics.* London: P.S King and Son.

[3] Coombs, C. (1964). *A theory of data.* New York: Wiley.

[4] Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7* (pp. 179-188).

[5] Kullback, S. (1959). *Information theory and Statistics.* New York: Wiley.

[6] Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

[7] Mitchell, T. (1997). *Machine Learning.* McGraw-Hill.

[8] Mitra, S. and Kaiser, J. (Eds.) (1993). *Handbook for Digital Signal Processing.* New York: Wiley.

[9] Cowan, G. (1998). *Statistical Data Analysis.* Oxford: Clarendon Press.

[10] Weiss, S. and Indurkhya, N. (1998). *Predictive Data Mining: A Practical Guide.* San Francisco: Morgan Kaufmann Publishers Inc.

[11] Rumelhart, D., McClelland, J. and the PDP Research Group, (Eds.) (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Vols I and II.* Cambridge, Mass.: MIT Press.

[12] Holland, J. (1992). Genetic algorithms. *Scientific American, 267,* No. 1.

[13] Thornton, C. (2000). *Truth from Trash: How Learning Makes Sense.* MIT Press.

[14] Thornton, C. (1999). What do constructive learners really learn?. *Artificial Intelligence Review, 13,* No. 4 (pp. 249-257).

[15] Thornton, C. and du Boulay, B. (1992). *Artificial Intelligence Through Search.* Intellect Press.

[16] Quinlan, J. (1993). *C4.5: Programs for Machine Learning.* San Mateo, California: Morgan Kaufmann.

[17] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees.* Wadsworth.

[18] Cover, M. and Hart, P. (1967). Nearest neighbr pattern classification. *IEEE Trans. Info. Theory, 13* (pp. 21-27).